IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION

PANTHER INNOVATIONS, LLC,
 	
      Plaintiff,
 	
v.
 	
MICROSOFT CORPORATION,
 	
      Defendant.
	CIVIL ACTION NO. 6:20-cv-1071
	
	JURY TRIAL DEMANDED

PLAINTIFF'S COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Panther Innovations, LLC ("Panther" or "Plaintiff") files this Complaint for patent infringement against Microsoft Corporation ("Microsoft" or "Defendant") and states as follows:

NATURE OF THE ACTION

1.   This is a civil action for patent infringement under the patent laws of the United States of America, 35 U.S.C. § 1 *et seq*.

2.   Panther is the owner of all right, title, and interest in U.S. Patent Nos. 7,840,652 (the "'652 Patent") and 8,069,231 (the "'231 Patent") (collectively "the Asserted Patents"), which are attached as Exhibits A and B and incorporated herein by reference.

3.   Defendant Microsoft has infringed and continues to infringe one or more claims of the Asserted Patents by making, using, offering to sell, and selling within the United States, including in this District, certain products and services. Panther seeks to recover monetary damages, attorneys' fees, and costs.

THE PARTIES

4.   Panther is a Texas limited liability company with a principal place of business at 2325 Oak Alley, Tyler, Texas 75703.

5.   Defendant Microsoft is a corporation organized under the laws of the State of Washington, with its principal place of business at 1 Microsoft Way, Redmond, Washington 98052. Defendant may be served via its registered agent, Corporation Service Company, at 211 East 7th Street, Suite 620, Austin, Texas 78701.

6.   Microsoft has been registered to do business in the state of Texas under Texas SOS file number 0006776606 since at least 1986.

JURISDICTION AND VENUE

7.   The Court has subject matter jurisdiction over this action pursuant to 28 U.S.C. §§ 1331 and 1338(a).

8.   This Court has personal jurisdiction over Defendant because Defendant does continuous and systematic business in this District, including by providing infringing products and services to the residents of the Western District of Texas that Defendant knew would be used within this District, and by soliciting business from the residents of the Western District of Texas. For example, Defendant is subject to personal jurisdiction in this Court because, *inter alia*, Defendant has regular places of business in the District at 10900 Stonelake Boulevard, Suite 225, Austin, Texas 78759 and Concord Park II, 401 East Sonterra Boulevard, Suite 300, San Antonio, Texas 78258. Defendant directly, and through agents, regularly does, solicits, and transacts business in the Western District of Texas.

9.   Defendant has committed and continues to commit acts of infringement in violation of 35 U.S.C. § 271 within the Western District of Texas. Defendant has in the past made, used,

marketed, distributed, offered for sale, sold, and/or imported infringing products, and performed infringing methods, in the State of Texas and in the Western District of Texas.   Defendant continues to make, use, market, distribute, offer for sale, sell, and/or import infringing products, and perform infringing methods, in the State of Texas and in the Western District of Texas. Accordingly, Defendant has in the past engaged, and continues to engage, in infringing conduct within and directed at or from this District.   Additionally, Defendant has purposefully and voluntarily placed its infringing products into the stream of commerce with the expectation that its infringing products will be used in this District. The infringing products have been and continue to be distributed to and used in this District. Upon information and belief, the infringing products have been and continue to be distributed from this District. Defendant's acts have caused, and continue to cause, injury to Plaintiff, including within this District.

10.   Venue is proper in this District under the provisions of 28 U.S.C. §§ 1391 and 1400(b) at least because Defendant has committed acts of infringement in this District and has a regular and established places of business in this District at 10900 Stonelake Boulevard, Suite 225, Austin, Texas 78759 and Concord Park II, 401 East Sonterra Boulevard, Suite 300, San Antonio, Texas 78258.

BACKGROUND

11.   Years before Microsoft added the accused functionality to its Windows operating system, Ascentive, LLC was developing intellectual property directed to optimizing network speeds. Adam Schran and Robert Darlington, the named inventors of the Asserted Patents, sought to address the inefficiency in the Internet connection by computers and the difficulty of adjusting network configuration settings of a computer for Internet data transfer. Mr. Schran and Mr. Darlington conceived of a novel way of optimizing a computer's Internet connection by selecting

from groups of network configuration settings and conducting one or more performance tests to automatically adjust the configuration settings to enhance the end-user's Internet performance. This invention resulted in the Asserted Patents.

U.S. PATENT NO. 7,840,652

12.   On October 15, 2004, the United States Patent and Trademark Office duly and legally issued the '652 Patent, entitled "System and method for determining network configuration settings that provide optimal network performance" after a full and fair examination.

13.   Exhibit A is a true and correct copy of the '652 Patent.

14.   The '652 Patent is valid and enforceable under United States patent laws.

15.   Plaintiff is the owner of the '652 Patent, having received all right, title and interest in and to the '652 Patent from the previous assignee of record.

16.   Plaintiff possesses all rights of recovery under the '652 Patent, including the exclusive right to recover for past infringement.

U.S. PATENT NO. 8,069,231

17.   On November 29, 2011, the United States Patent and Trademark Office duly and legally issued the '231 Patent, entitled "Computer program product for determining a group of network configuration settings that provide optimal network performance" after a full and fair examination.

18.   Exhibit B is a true and correct copy of the '231 Patent.

19.   The '231 Patent is valid and enforceable under United States patent laws.

20.   Plaintiff is the owner of the '231 Patent, having received all right, title and interest in and to the '231 Patent from the previous assignee of record.

21.   Plaintiff possesses all rights of recovery under the '231 Patent, including the exclusive right to recover for past infringement.

THE ASSERTED PATENTS

22.   The claims of the Asserted Patents are directed to patent-eligible, non-abstract inventions.

23.   The Asserted Patents address, among other things, specific technological improvements for optimizing Internet data transfer speeds. For example, when a computer connects to a server across a TCP/IP network, such as using a home computer to connect to www.bing.com, that computer's operating system connects using certain user-selectable pre-configured TCP/IP settings. However, optimal settings for connecting to that server or any other server are not static; optimal settings for connections to that server will vary across time based on certain factors such as congestion and server demand. *See, e.g.*, Ex. A '652 Patent at 1:30-50. The Asserted Patents are directed to optimizing that connection by automatically adjusting specific user-selectable TCP/IP network settings by selecting from groups of network configuration settings and conducting one or more performance test to ensure the best data throughput. *See, e.g.*, *id*. at 1:59-2:55. These settings can also be continually tested to ensure an optimal connection in varying situations. *Id.*

24.   For instance, the Asserted Patents describe certain TCP/IP settings that can be adjusted to optimize the network connection, including, for example, Maximum Transmission Unit (MTU), Maximum Segment Size (MSS), Receive Window (RWIN), Time to Live (TTL), Black Hole Detection, and MTU Auto Discovery. *See, e.g.*, *id*. at 2:41-46.

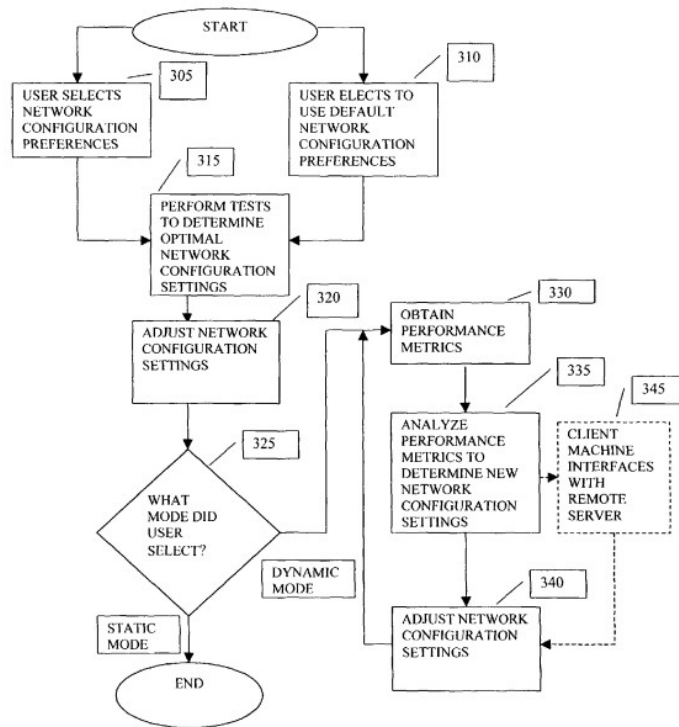25.   Figure 3 of the Asserted Patents is illustrated below:

Fig. 3

26.   Figure 3 generally describes a process where a user may either select network configuration preferences (305) or elect to use default network configuration settings (310). *See, e.g., id.* at 5:18-7:22. An application, for example, then performs tests to determine an optimal network configuration (315). *Id.* A dynamic mode (330, 335, and 340) is also described where the network settings are continuously analyzed and adjusted to ensure optimal network connectivity. *Id.*

MICROSOFT'S INFRINGING PRODUCTS AND ACTIVITIES

27.   Microsoft Windows Products (the "Accused Products") provide various user-selectable network configuration settings. The Accused Products include at least Windows Vista, Windows 7, Windows 8, and Windows 10 operating systems, including at least as made available as software for download or purchase, software pre-installed in Microsoft or OEM computer products, and software pre-installed in Microsoft Xbox and Surface products. Upon information

and belief, the accused features were introduced in 2008 in the Windows Vista and Windows

Server 2008 operating systems with, for example, user-selectable TCP Auto-tuning functionality,

user-selectable Congestion Provider functionality, and user-selectable Scaling Heuristics

functionality.

28.   For example, Microsoft Windows Products provide TCP auto-tuning. A TCP auto-

tuning level of normal will provide a scale factor of 8 (which is the maximum scale factor to be

used) to automatically optimize RWIN. The autotuning feature implements a sliding window

based on network constraints:

Autotuning levels

You can set receive window autotuning to any of five levels. The default level is **Normal**. The following table describes the levels.

Level	Hexadecimal value	Comments
Normal (default)	0x8 (scale factor of 8)	Set the TCP receive window to grow to accommodate almost all scenarios.
Disabled	No scale factor available	Set the TCP receive window at its default value.
Restricted	0x4 (scale factor of 4)	Set the TCP receive window to grow beyond its default value, but limit such growth in some scenarios.
Highly Restricted	0x2 (scale factor of 2)	Set the TCP receive window to grow beyond its default value, but do so very conservatively.
Experimental	0xE (scale factor of 14)	Set the TCP receive window to grow to accommodate extreme scenarios.

https://docs.microsoft.com/en-us/windows-server/networking/technologies/network-subsystem/net-sub-performance-tuning-nics (last visited Nov. 18, 2020).

How the Receive Window Auto-Tuning feature improves data transfer

The Receive Window Auto-Tuning feature lets the operating system continually monitor routing conditions such as bandwidth, network delay, and application delay. Therefore, the operating system can configure connections by scaling the TCP receive window to maximize the network performance. To determine the optimal receive window size, the Receive Window Auto-Tuning feature measures the products that delay bandwidth and the application retrieve rates. Then, the Receive Window Auto-Tuning feature adapts the receive window size of the ongoing transmission to take advantage of any unused bandwidth.

https://support.microsoft.com/en-us/help/947239/description-of-the-receive-window-auto-tuning-feature-for-http-traffic (last visited Nov. 18, 2020).

29.   On information and belief, Microsoft Windows Products are enabled to implement Congestion Provider functionality to automatically optimize network configuration settings including MSS. By default, client computers use NewReno, but may also be configured to implement CTCP or DCTCP:

-CongestionProvider

Specifies the congestion provider property that TCP uses. The acceptable values for this parameter are:

- CTCP. Compound TCP increases the receive window and amount of data sent. CTCP can improve throughput on higher latency connections.
- DCTCP. Data Center TCP adjusts the TCP window based on network congestion feedback based on Explicit Congestion Notification (ECN) signaling. DCTCP may improve throughput on low latency links.
- Default. Servers use DCTCP by default. Client computers use NewReno. For information about NewReno, see RFC 3782.

Type:	CongestionProvider
Accepted values:	Default, CTCP, DCTCP
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

https://docs.microsoft.com/en-us/powershell/module/nettcpip/set-nettcpsetting?view=win10-ps (last visited Nov. 18, 2020).

30.   CTCP, or Compound TCP, is a Microsoft implementation that includes the ability to automatically optimize performance:

Many approaches have been proposed to improve the connection's throughput by adopting more aggressive loss-based CC algorithms. These algorithms, although can effectively improve the link utilization, have the weakness of poor RTT fairness. Further, they may severely decrease the performance of regular TCP flows that traverse the same network path. On the other hand, pure delay-based approaches that improve the throughput in high-speed network may not work well under the environment where the background traffics are mixed with both delay-based and greedy loss-based flows.

We propose a novel Compound TCP (CTCP) approach, which is a synergy of delay-based and the loss-based approach. In CTCP, we add a scalable delay-based component into the standard TCP Reno congestion avoidance algorithm (i.e., the loss-based component). The sending rate of CTCP is controlled by both two components. Augmented with this new delay-based component, which rapidly increases sending rate when network path is under utilized but gracefully retreats in a busy network when bottleneck queue is built, CTCP has very good bandwidth scalability with improved RTT fairness, and at the same time achieves good TCP-fairness, irrelevant to the windows size.

The novel Compound TCP has the following properties:

1. CTCP can efficiently use the network resource and achieve high link utilization.
2. CTCP has the similar or even improved RTT fairness regarding to regular TCP.
3. CTCP keeps good TCP-friendliness.

Compound TCP has been shipped in Microsoft Windows Vista and later Operating Systems.

https://www.microsoft.com/en-us/research/project/the-compound-tcp-for-high-speed-and-long-distance-networks/; *see also*, https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2005-86.pdf (last visited Nov. 18, 2020).

31.   The Accused Products also are enabled to implement Scaling Heuristics functionality to optimize scaling:

-ScalingHeuristics

Specifies whether to enable scaling heuristics. The acceptable values for this parameter are:

- Enabled
- Disabled

Type:	ScalingHeuristics
Accepted values:	Disabled, Enabled
Position:	Named
Default value:	None
Accept pipeline input:	False
Accept wildcard characters:	False

https://docs.microsoft.com/en-us/powershell/module/nettcpip/set-nettcpsetting?view=win10-ps (last visited Nov. 18, 2020)

> This policy setting allows you to configure Window Scaling Heuristics. Window Scaling Heuristics is an algorithm to identify connectivity and throughput problems caused by many Firewalls and other middle boxes that don't interpret Window Scaling option correctly.If you do not configure this policy setting the local host settings are used.If you enable this policy setting Window Scaling Heuristics will be enabled and system will try to identify connectivity and throughput problems and take appropriate measures.If you disable this policy setting Window Scaling Heuristics will be disabled and system will not try to identify connectivity and throughput problems casued by Firewalls or other middle boxes.
>
> **Policy path:**
> Network\TCPIP Settings\Parameters
> **Scope:**
> Machine
> **Supported on:**
> At least Windows Vista Service Pack 1
> **Registry settings:**
> HKLM\System\CurrentControlSet\Services\Tcpip\Parameters!EnableWsd
> **Filename:**
> tcpip.admx

https://www.windows-security.org/192ee4e61e5b3a5334794f4af06f03a5/set-window-scaling-heuristics-state (last visited Nov. 18, 2020).

32.   For each of the TCP auto-tune, Congestion Provider, and Scaling Heuristics functionalities, the Accused Products are designed to automatically adjust these network configuration settings to optimize connectivity. For example, for TCP auto-tune functionality, the Accused Products conduct performance tests to optimize the network feature:

> **How the Receive Window Auto-Tuning feature improves data transfer**
>
> The Receive Window Auto-Tuning feature lets the operating system continually monitor routing conditions such as bandwidth, network delay, and application delay. Therefore, the operating system can configure connections by scaling the TCP receive window to maximize the network performance. To determine the optimal receive window size, the Receive Window Auto-Tuning feature measures the products that delay bandwidth and the application retrieve rates. Then, the Receive Window Auto-Tuning feature adapts the receive window size of the ongoing transmission to take advantage of any unused bandwidth.

https://support.microsoft.com/en-us/help/947239/description-of-the-receive-window-auto-tuning-feature-for-http-traffic (last visited Nov. 18, 2020).

33.   For the Congestion Provider functionality, the Accused Products conduct performance tests to optimize the network feature:

Many approaches have been proposed to improve the connection's throughput by adopting more aggressive loss-based CC algorithms. These algorithms, although can effectively improve the link utilization, have the weakness of poor RTT fairness. Further, they may severely decrease the performance of regular TCP flows that traverse the same network path. On the other hand, pure delay-based approaches that improve the throughput in high-speed network may not work well under the environment where the background traffics are mixed with both delay-based and greedy loss-based flows.

We propose a novel Compound TCP (CTCP) approach, which is a synergy of delay-based and the loss-based approach. In CTCP, we add a scalable delay-based component into the standard TCP Reno congestion avoidance algorithm (i.e., the loss-based component). The sending rate of CTCP is controlled by both two components. Augmented with this new delay-based component, which rapidly increases sending rate when network path is under utilized but gracefully retreats in a busy network when bottleneck queue is built, CTCP has very good bandwidth scalability with improved RTT fairness, and at the same time achieves good TCP-fairness, irrelevant to the windows size.

The novel Compound TCP has the following properties:

1. CTCP can efficiently use the network resource and achieve high link utilization.
2. CTCP has the similar or even improved RTT fairness regarding to regular TCP.
3. CTCP keeps good TCP-friendliness.

Compound TCP has been shipped in Microsoft Windows Vista and later Operating Systems.

https://www.microsoft.com/en-us/research/project/the-compound-tcp-for-high-speed-and-long-distance-networks/; *see also*, https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2005-86.pdf (last visited Nov. 18, 2020).

34.   For Scaling Heuristics, the Accused Products conduct performance tests to optimize the network feature:

This policy setting allows you to configure Window Scaling Heuristics. Window Scaling Heuristics is an algorithm to identify connectivity and throughput problems caused by many Firewalls and other middle boxes that don't interpret Window Scaling option correctly.If you do not configure this policy setting the local host settings are used.If you enable this policy setting Window Scaling Heuristics will be enabled and system will try to identify connectivity and throughput problems and take appropriate measures.If you disable this policy setting Window Scaling Heuristics will be disabled and system will not try to identify connectivity and throughput problems casued by Firewalls or other middle boxes.

Policy path:
Network\TCPIP Settings\Parameters
Scope:
Machine
Supported on:
At least Windows Vista Service Pack 1
Registry settings:
HKLM\System\CurrentControlSet\Services\Tcpip\Parameters!EnableWsd
Filename:
tcpip.admx

https://www.windows-security.org/192ee4e61e5b3a5334794f4af06f03a5/set-window-scaling-heuristics-state (last visited Nov. 18, 2020).

35.   Windows 10 is "active on more than 900 million devices."

https://www.microsoft.com/investor/reports/ar19/index.html (last visited Nov. 18, 2020).   The

Accused Products are sold to consumers and Microsoft's Original Equipment Manufacturers. *See*, *e.g.*, *id*. The Accused Products are also preinstalled on hardware devices sold by Microsoft including, at least, Xbox series devices and Surface series devices. *See*, *e.g.*, https://www.polygon.com/2015/8/10/9126899/phil-spencer-obliterating-distinction-between-xbox-pc-gamescom-interview-2015; https://www.quora.com/What-is-the-OS-of-a-Xbox-one; and https://support.microsoft.com/en-us/help/2858199/surface-supported-operating-systems. (last visited Nov. 18, 2020).

COUNT I: INFRINGEMENT OF U.S. PATENT NO. 7,840,652

36.   Plaintiff incorporates by reference and re-alleges the foregoing paragraphs of this Complaint as if fully set forth herein.

37.   Defendant has directly infringed and continues to directly infringe at least claims 1 and 29 of the '652 Patent in violation of 35 U.S.C. § 271 *et seq.*, by making, using, offering for sale, or selling in the United States, and/or importing into the United States without authority or license the Accused Products.

38.   The Accused Products meet all the limitations of at least claims 1 and 29 of the '652 Patent. For example, claim 1 of the '652 Patent recites:

> A method of optimizing network configuration settings for a user's client machine, the method comprising:
>
> (a) providing a plurality of groups of network configuration settings to be used by the user's client machine;
>
> (b) establishing a network connection between the user's client machine and a remote server;
>
> (c) selecting one of the groups of network configuration settings to be used by the user's client machine from the provided groups of settings, wherein step (c) is initiated on the user's client machine;
>
> (d) automatically conducting one or more performance tests using the selected network configuration settings during the established network connection;

(e) repeating steps (c) and (d) for one or more other groups of network configuration settings during the established network connection; and

(f) automatically adjusting the network configuration settings of the user's client machine provided in the groups based on the results of the performance tests, wherein the adjusted network configuration settings are settings that optimize the performance of the user's client machine.

39.   A non-limiting and exemplary claim chart comparing the Accused Products to claims 1 and 29 of the '652 Patent is attached hereto as Exhibit C and is incorporated herein as if fully rewritten. This description is based on publicly available information. Plaintiff reserves the right to modify this description, including, for example, on the basis of information about the Accused Products that it obtains during discovery.

40.   As in claim 1 of the '652 Patent, the Accused Products optimize network configuration settings for a user's client machine (*e.g.*, a computer running one of the MS Windows Products that is an Accused Product).

41.   As in claim 1 of the '652 Patent, the Accused Products provide a plurality of groups of network configuration settings (*e.g.*, TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality) to be used by the user's client machine. For example, a TCP auto-tuning level of normal will provide a scale factor of 8 (which is the maximum scale factor to be used) to automatically optimize RWIN. The auto-tuning feature implements a sliding window based on network constraints. *See*, *e.g.*, https://docs.microsoft.com/en-us/windows-server/networking/technologies/network-subsystem/net-sub-performance-tuning-nics; https://support.microsoft.com/en-us/help/947239/description-of-the-receive-window-auto-tuning-feature-for-http-traffic (last visited Nov. 18, 2020). In addition, the Accused Products implement Congestion Provider functionality to automatically optimize network settings including MSS. By default, client computers use NewReno, but may also be configured to implement CTCP or   DCTCP.   *See*,   *e.g.*,   https://docs.microsoft.com/en-us/powershell/module/nettcpip/set-

nettcpsetting?view=win10-ps (last visited Nov. 18, 2020). CTCP, or Compound TCP, is a

Microsoft implementation that includes the ability to automatically optimize performance. *See*,

e.g.,   https://www.microsoft.com/en-us/research/project/the-compound-tcp-for-high-speed-and-

long-distance-networks/;   https://www.microsoft.com/en-us/research/wp-

content/uploads/2016/02/tr-2005-86.pdf (last visited Nov. 18, 2020). The Accused Products also

implement   Scaling   Heuristics   functionality   to   optimize   scaling.   *See*,   *e.g.*,

https://docs.microsoft.com/en-us/powershell/module/nettcpip/set-nettcpsetting?view=win10-ps;

https://www.windows-security.org/192ee4e61e5b3a5334794f4af06f03a5/set-window-scaling-

heuristics-state (last visited Nov. 18, 2020).

42.   As in claim 1 of the '652 Patent, the Accused Products establish a network

connection between the user's client machine (*e.g.*, a computer running one of the Accused

Products) and a remote server (*e.g.*, a webserver). For example, the Accused Products are designed

with certain functionality that is specifically directed to establishing a network connection between

a client machine and remote servers using TCP. WinHTTP is one such example that is designed

to allow a computer running one of the Accused Products to communicate with a remote server.

See, *e.g.*,   https://support.microsoft.com/en-us/help/947239/description-of-the-receive-window-

auto-tuning-feature-for-http-traffic (last visited Nov. 18, 2020).

43.   As in claim 1 of the '652 Patent, the Accused Products select one of the groups of

network configurations settings (*e.g.*, one of TCP auto-tune, Congestion Provider, and Scaling

Heuristics functionality) to be used by the user's client machine from the provided groups of

settings (*e.g.*, TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality), where

this step is initiated on the client machine. For example, the Accused Products implement separate

unique algorithms that select from and adjust network configuration settings, such as TCP auto-

tune, Congestion Provider, and Scaling Heuristics functionality. *See, e.g.,* https://docs.microsoft.com/en-us/windows-server/networking/technologies/network-subsystem/net-sub-performance-tuning-nics#autotuning-levels (last visited Nov. 18, 2020).

44.   As in claim 1 of the '652 Patent, the Accused Products automatically conduct one or more performance tests using the selected network configuration settings (*e.g.,* one of TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality) during the established network connection.   For example, for TCP auto-tune functionality, the Accused Products conduct performance tests to optimize the network feature. *See, e.g.,* https://support.microsoft.com/en-us/help/947239/description-of-the-receive-window-auto-tuning-feature-for-http-traffic   (last visited Nov. 18, 2020). Similarly, for the Congestion Provider functionality, the Accused Products conduct performance tests to optimize the network feature. *See, e.g.,* https://www.microsoft.com/en-us/research/project/the-compound-tcp-for-high-speed-and-long-distance-networks/;   https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2005-86.pdf (last visited Nov. 18, 2020). Further, for Scaling Heuristics, the Accused Products conduct performance tests to optimize the network feature. *See, e.g.,* https://www.windows-security.org/192ee4e61e5b3a5334794f4af06f03a5/set-window-scaling-heuristics-state   (last visited Nov. 18, 2020).

45.   As in claim 1 of the '652 Patent, the Accused Products repeat steps (c) and (d) for one or more other groups of network configuration settings (*e.g.,* one of TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality) during the established network connection.   For example, the Accused Products optimize each of TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality.

46.   As in claim 1 of the '652 Patent, the Accused Products automatically adjust the network configuration settings (*e.g.*, TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality) of the user's client machine (*e.g.*, a computer running MS Windows Products) provided in the groups based on the results of the performance tests, wherein the adjusted network configuration settings are settings that optimize the performance of the user's client machine. For example, the purpose of TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality is   to   optimize   the   performance.   *See*,   *e.g.*,   https://support.microsoft.com/en-us/help/947239/description-of-the-receive-window-auto-tuning-feature-for-http-traffic; https://www.microsoft.com/en-us/research/project/the-compound-tcp-for-high-speed-and-long-distance-networks/;   https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2005-86.pdf;   https://www.windows-security.org/192ee4e61e5b3a5334794f4af06f03a5/set-window-scaling-heuristics-state (last visited Nov. 18, 2020).

47.   Defendant makes, uses, sells, and/or offers to sell the Accused Products which practice at least claim 1 of the '652 Patent and which comprise all of the elements of claim 29 of the '652 Patent.

48.   In violation of 35 U.S.C. § 271, Defendant is now, and has been directly infringing the '652 Patent, including through its own use, testing, and sale of the Accused Products.

49.   Defendant has had knowledge of infringement of the '652 Patent at least as of the service of the present Complaint.

50.   Defendant has directly infringed and continues to directly infringe at least one claim of the '652 Patent by making, using, offering for sale, and selling the Accused Products without authority in the United States. As a direct and proximate result of Defendant's direct infringement of the '652 Patent, Panther has been and continues to be damaged.

51.   By engaging in the conduct described herein, Defendant has injured Panther and is thus liable for infringement of the '652 Patent, pursuant to 35 U.S.C. § 271.

52.   Defendant has committed these acts of infringement without license or authorization.

53.   As a result of Defendant's infringement of the '652 Patent, Plaintiff has suffered monetary damages and is entitled to a monetary judgment in an amount adequate to compensate for Defendant's past infringement, together with interests and costs.

54.   Plaintiff reserves the right to modify its infringement theories as discovery progresses in this case; it shall not be estopped for infringement contention or claim construction purposes by the claim chart that it provides with this Complaint. The claim chart depicted in Exhibit C is intended to satisfy the notice requirements of Rule 8(a)(2) of the Federal Rule of Civil Procedure and does not represent Plaintiff's preliminary or final infringement contentions or preliminary or final claim construction positions.

COUNT II: INFRINGEMENT OF U.S. PATENT NO. 8,069,231

55.   Plaintiff incorporates by reference and re-alleges the foregoing paragraphs of this Complaint as if fully set forth herein.

56.   Defendant has directly infringed and continues to infringe at least claim 1 of the '231 Patent in violation of 35 U.S.C. § 271 *et seq.*, by making, using, offering for sale, or selling in the United States, and/or importing into the United States without authority or license the Accused Product.

57.   The Accused Products meet all the limitations of at least claim 1 of the '231 Patent. For example, claim 1 of the '231 Patent recites:

> A computer program product for optimizing network configuration settings for a
> user's client machine, the computer program product comprising non-transitory

17

computer-readable media encoded with instructions for execution by a processor to perform a method comprising:

(a) providing a plurality of groups of network configuration settings to be used by the user's client machine;

(b) establishing a network connection between the user's client machine and a remote server;

(c) selecting one of the groups of network configuration settings to be used by the user's client machine from the provided groups of settings, wherein step (c) is initiated on the user's client machine;

(d) automatically conducting one or more performance tests using the selected network configuration settings during the established network connection;

(e) repeating steps (c) and (d) for one or more other groups of network configuration settings during the established network connection; and

(f) automatically adjusting the network configuration settings of the user's client machine provided in the groups based on the results of the performance tests, wherein the adjusted network configuration settings are settings that optimize the performance of the user's client machine.

58.   A non-limiting and exemplary claim chart comparing the Accused Products to claim 1 of the '231 Patent is attached hereto as Exhibit C and is incorporated herein as if fully rewritten. This description is based on publicly available information. Plaintiff reserves the right to modify this description, including, for example, on the basis of information about the Accused Product that it obtains during discovery.

59.   As in claim 1 of the '231 Patent, the Accused Products are computer program products for optimizing network configuration settings for a user's client machine (*e.g.*, a computer running one the MS Windows Products that is an Accused Product), the Accused Products include non-transitory computer-readable media (*e.g.*, software) encoded with instructions for execution by a processor to perform the claimed function.

60.   As in claim 1 of the '231 Patent, the Accused Products are designed to enable providing a plurality of groups of network configuration settings (*e.g.*, TCP auto-tune, Congestion

Provider, and Scaling Heuristics functionality) to be used by the user's client machine. For example, a TCP auto-tuning level of normal will provide a scale factor of 8 (which is the maximum scale factor to be used) to automatically optimize RWIN. The auto-tuning feature implements a sliding window based on network constraints. *See*, *e.g.*, https://docs.microsoft.com/en-us/windows-server/networking/technologies/network-subsystem/net-sub-performance-tuning-nics; https://support.microsoft.com/en-us/help/947239/description-of-the-receive-window-auto-tuning-feature-for-http-traffic (last visited Nov. 18, 2020). In addition, the Accused Products are enabled to implement Congestion Provider functionality to automatically optimize network settings including MSS. By default, client computers use NewReno, but may also be configured to implement CTCP or DCTCP. *See*, *e.g.*, https://docs.microsoft.com/en-us/powershell/module/nettcpip/set-nettcpsetting?view=win10-ps (last visited Nov. 18, 2020). CTCP, or Compound TCP, is a Microsoft implementation that includes the ability to automatically optimize performance. *See*, *e.g.*, https://www.microsoft.com/en-us/research/project/the-compound-tcp-for-high-speed-and-long-distance-networks/; https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2005-86.pdf (last visited Nov. 18, 2020). The Accused Products also are enabled to implement Scaling Heuristics functionality to optimize scaling. *See*, *e.g.*, https://docs.microsoft.com/en-us/powershell/module/nettcpip/set-nettcpsetting?view=win10-ps; https://www.windows-security.org/192ee4e61e5b3a5334794f4af06f03a5/set-window-scaling-heuristics-state (last visited Nov. 18, 2020).

61.   As in claim 1 of the '231 Patent, the Accused Products are designed to enable establishing a network connection between the user's client machine (*e.g.*, a computer running one of the MS Windows Products that is an Accused Product) and a remote server (*e.g.*, a webserver).

For example, the Accused Products are designed with certain functionality that is specifically directed to establishing a network connection between a client machine and remote servers using TCP. WinHTTP is one such example that is designed to allow a computer running one of the Accused Products to communicate with a remote server. *See*, *e.g.*, https://support.microsoft.com/en-us/help/947239/description-of-the-receive-window-auto-tuning-feature-for-http-traffic (last visited Nov. 18, 2020).

62.   As in claim 1 of the '231 Patent, the Accused Products are designed to enable selecting one of the groups of network configurations settings (*e.g.*, one of TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality) to be used by the user's client machine from the provided groups of settings (*e.g.*, TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality), where this step is initiated on the client machine. For example, the Accused Products are designed with separate unique algorithms that are designed to select from and adjust network configuration settings, such as TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality. *See*, *e.g.*, https://docs.microsoft.com/en-us/windows-server/networking/technologies/network-subsystem/net-sub-performance-tuning-nics#autotuning-levels (last visited Nov. 18, 2020).

63.   As in claim 1 of the '231 Patent, the Accused Products are designed to enable automatically conducting one or more performance tests using the selected network configuration settings (*e.g.*, one of TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality) during the established network connection. For example, for TCP auto-tune functionality, the Accused Products conduct performance tests to optimize the network feature. *See*, *e.g.*, https://support.microsoft.com/en-us/help/947239/description-of-the-receive-window-auto-tuning-feature-for-http-traffic (last visited Nov. 18, 2020). Similarly, for the Congestion Provider

functionality, the Accused Products conduct performance tests to optimize the network feature. *See*, *e.g.*, https://www.microsoft.com/en-us/research/project/the-compound-tcp-for-high-speed-and-long-distance-networks/; https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2005-86.pdf (last visited Nov. 18, 2020). Further, for Scaling Heuristics, the Accused Products conduct performance tests to optimize the network feature. *See*, *e.g.*, https://www.windows-security.org/192ee4e61e5b3a5334794f4af06f03a5/set-window-scaling-heuristics-state (last visited Nov. 18, 2020).

64.   As in claim 1 of the '231 Patent, the Accused Products are designed to enable repeating steps (c) and (d) for one or more other groups of network configuration settings (*e.g.*, one of TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality) during the established network connection. For example, the Accused Products optimize each of TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality.

65.   As in claim 1 of the '231 Patent, the Accused Products are designed to enable automatically adjusting the network configuration settings (*e.g.*, TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality) of the user's client machine (*e.g.*, a computer running one of the MS Windows Products that is an Accused Product) provided in the groups based on the results of the performance tests, wherein the adjusted network configuration settings are settings that optimize the performance of the user's client machine. For example, the purpose of TCP auto-tune, Congestion Provider, and Scaling Heuristics functionality is to optimize the performance. *See*, *e.g.*, https://support.microsoft.com/en-us/help/947239/description-of-the-receive-window-auto-tuning-feature-for-http-traffic; https://www.microsoft.com/en-us/research/project/the-compound-tcp-for-high-speed-and-long-distance-networks/; https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-2005-86.pdf;

https://www.windows-security.org/192ee4e61e5b3a5334794f4af06f03a5/set-window-scaling-heuristics-state (last visited Nov. 18, 2020).

66.   Defendant makes, uses, sells, and/or offers to sell the Accused Products which practice at least claim 1 of the '231 Patent.

67.   In violation of 35 U.S.C. § 271, Defendant is now, and has been directly infringing the '231 Patent, including through its own use, testing, and sale of the Accused Products.

68.   Defendant has had knowledge of infringement of the '231 Patent at least as of the service of the present Complaint.

69.   Defendant has directly infringed and continues to directly infringe at least one claim of the '231 Patent by making, using, offering for sale, and selling the Accused Product without authority in the United States. As a direct and proximate result of Defendant's direct infringement of the '231 Patent, Panther has been and continues to be damaged.

70.   By engaging in the conduct described herein, Defendant has injured Panther and is thus liable for infringement of the '231 Patent, pursuant to 35 U.S.C. § 271.

71.   Defendant has committed these acts of infringement without license or authorization.

72.   As a result of Defendant's infringement of the '231 Patent, Plaintiff has suffered monetary damages and is entitled to a monetary judgment in an amount adequate to compensate for Defendant's past infringement, together with interests and costs.

73.   Plaintiff reserves the right to modify its infringement theories as discovery progresses in this case; it shall not be estopped for infringement contention or claim construction purposes by the claim chart that it provides with this Complaint. The claim chart depicted in Exhibit C is intended to satisfy the notice requirements of Rule 8(a)(2) of the Federal Rule of Civil

Procedure and does not represent Plaintiff's preliminary or final infringement contentions or preliminary or final claim construction positions.

DEMAND FOR JURY TRIAL

74.   Panther demands a trial by jury of any and all causes of action.

PRAYER FOR RELIEF

WHEREFORE, Panther respectfully requests:

a.   That Judgment be entered that Defendant has infringed one or more claims of the '652 Patent;

b.   That Judgment be entered that Defendant has infringed one or more claims of the '231 Patent;

c.   An award of damages pursuant to 35 U.S.C. §284, sufficient to compensate Plaintiff for the Defendant's past infringement and any continuing or future infringement;

d.   An assessment of pre-judgment and post-judgment interest and costs against Defendant, together with an award of such interest and costs, in accordance with 35 U.S.C. §284;

e.   That Defendant be directed to pay enhanced damages, including Plaintiff's attorneys' fees incurred in connection with this lawsuit pursuant to 35 U.S.C. §285; and

f.   That Plaintiff be granted such other and further relief as this Court may deem just and proper.

Dated: November 20, 2020   Respectfully submitted,

                                                      By: *Kevin S. Kudlac*
                                                      Kevin S. Kudlac
                                                      Texas Bar No. 00790089
                                                      kevin@connorkudlaclee.com
                                                      Cabrach J. Connor
                                                      Texas Bar No. 24036390
                                                      cab@connorkudlaclee.com

Jennifer Tatum Lee
Texas Bar No. 24046950
jennifer@connorkudlaclee.com

CONNOR KUDLAC LEE PLLC
609 Castle Ridge Road, Suite 450
Austin, Texas 78746
512.777.1254 Telephone
888.387.1134 Facsimile

Rodney R. Miller
Texas Bar No. 24070280
Admitted in this District
rmiller@hallboothsmith.com
HALL BOOTH SMITH, P.C.
191 Peachtree Street, NE, Suite 2900
Atlanta, GA 30303
T: 404.586.6601
F: 404.954.5020

Daniel C. Miller
New York Bar No. 4232773
Admitted in this District
dmiller@hallboothsmith.com
HALL BOOTH SMITH, P.C.
366 Madison Ave, Fifth Floor
New York, NY 10017
T: 917.805.2460

Karl Braun
Tennessee Bar No. 022371
(*Pro Hac Vice forthcoming*)
kbraun@hallboothsmith.com
HALL BOOTH SMITH, P.C.
424 Church, Suite 2950
Nashville, TN 37219
T: 615.313.9911

Attorneys for Plaintiff Panther Innovations, LLC